

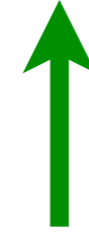
# GUI와 이벤트 구동 프로그래밍

## 1

# Graphical User Interface and Event-Driven Programming



컴퓨터와 사용자 사이의 소통 창구



# Graphical User Interface



GUI

문자가 아닌 눈에 보이는 다른 것들로 소통

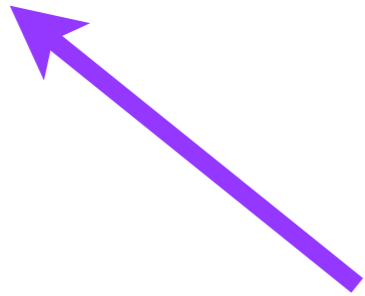
java.awt  
javax.swing

AWT = Abstract Window Toolkit

사용자 주도로  
수시로 발생하는 일

## Action Event

- 마우스 움직임
- 버튼 누름
- 메뉴 선택
- 키보드 입력



# Event-Driven

# Programming

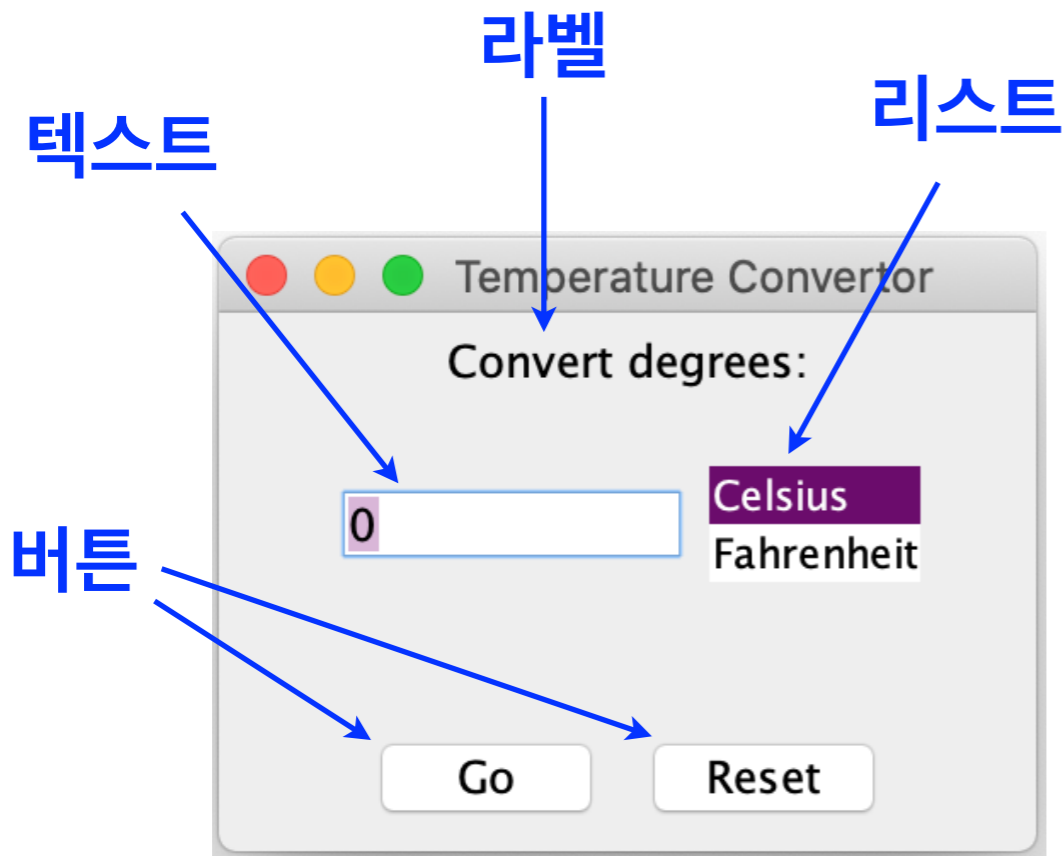
**Event-handler = Action Listener**

액션 이벤트가 발생하기를 기다리고 있다가,  
액션 이벤트가 발생하면 처리

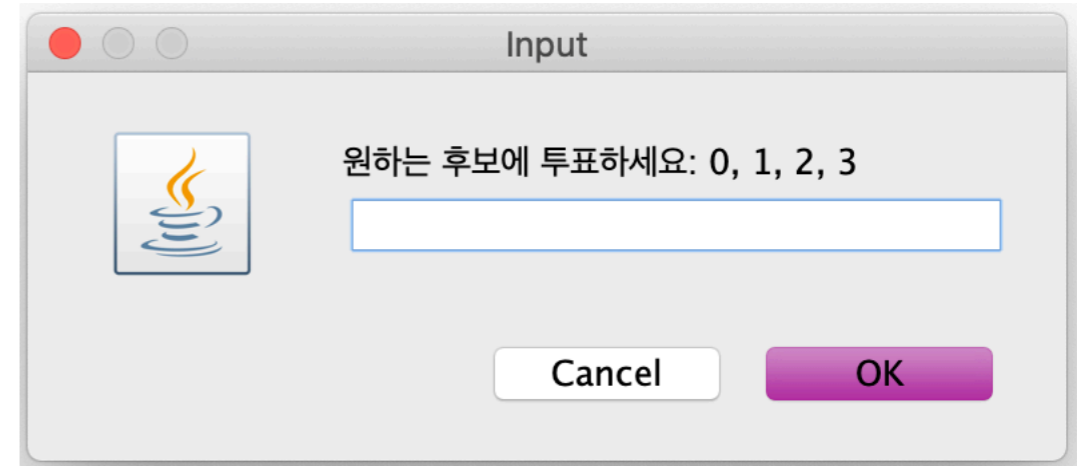
## The AWT/Swing Terminology

| 용어        |      | 기능  |
|-----------|------|---|
| component | 컴포넌트 | 이벤트가 발생하는 객체로서 크기와 위치가 있음<br>예: 라벨, 텍스트, 버튼, 리스트 등. |
| container | 컨테이너 | 다른 컴포넌트를 1개 이상 담을 수 있는 컴포넌트                         |
| panel     | 패널   | 표준 컨테이너   |
| window    | 윈도우  | 화면에 보이는 가장 바깥에 위치한 컨테이너로서<br>패널을 담고 있음              |
| frame     | 프레임  | 윈도우에 제목과 메뉴가 달린 틀                                   |
| dialog    | 대화창  | 대화를 위하여 띄우는 임시 창                                    |
| menu bar  | 메뉴바  | 선택할 메뉴가 있으며 프레임에 달려 있음                              |

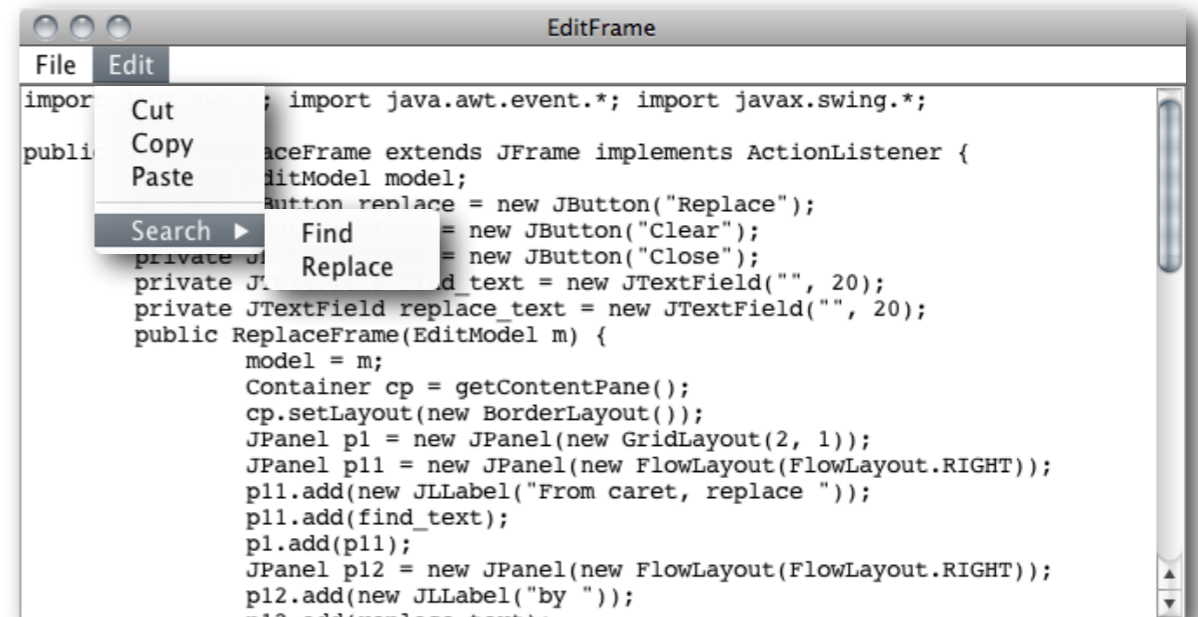
## 윈도우 프레임 window



## 대화창 dialog



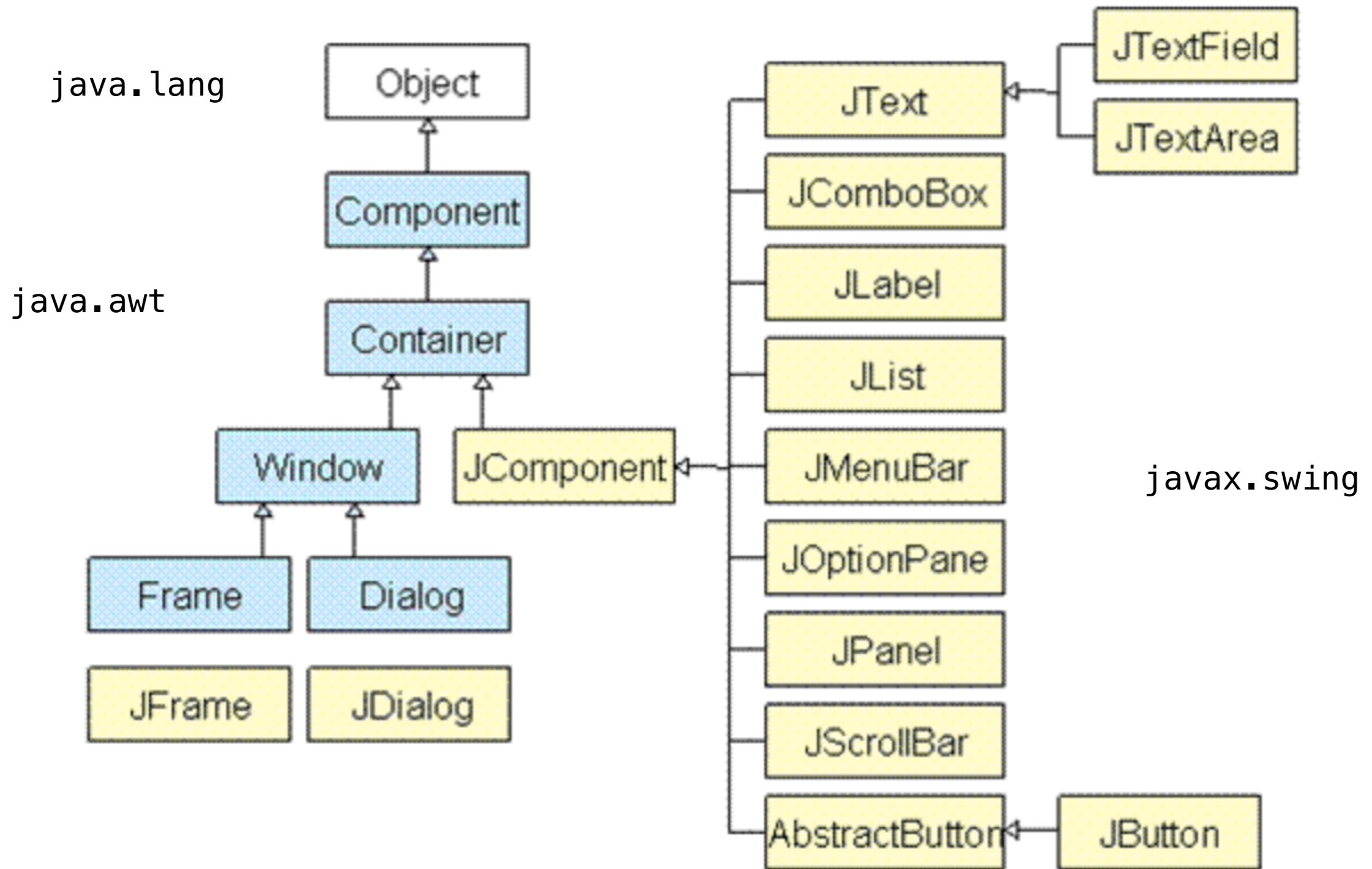
## 메뉴바 menu bar



## 자리 나누기 Layout

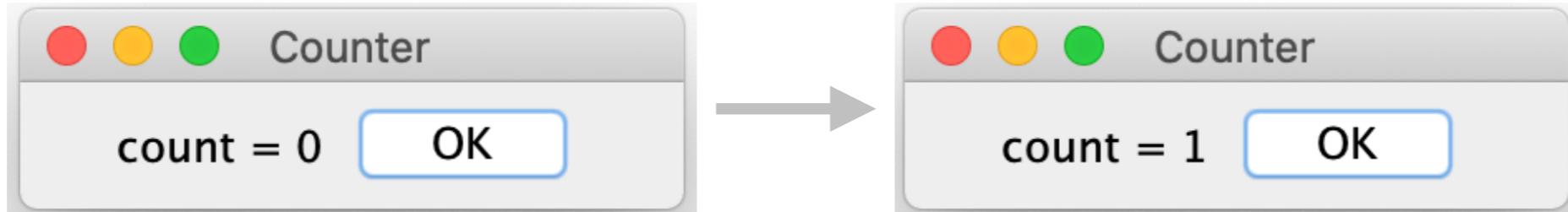
1. Flow layout : 나란히 가로로 나열
2. Border layout : 동, 서, 남, 북, 가운데로 나누어 자리 배정
3. Grid layout : 가로, 세로 균일한 크기로 바둑판 모양으로 자리 배정

# The AWT/Swing Class Hierarchy



# 사례 학습 #1 : 카운터 만들기

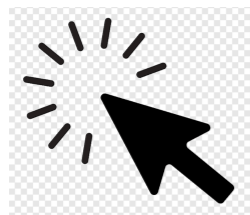
버튼을 누를 때마다 1씩 증가하는 카운터 만들기



```
public interface ActionListener {  
    /** actionPerformed - 액션 이벤트를 처리  
     * @param e - 발생한 이벤트 관련 정보 */  
    public void actionPerformed(ActionEvent e);  
}
```

java.awt.event

```
public class C implements ActionListener {  
    . . .  
    public void actionPerformed(ActionEvent e) {  
        . . .  
    }  
}
```

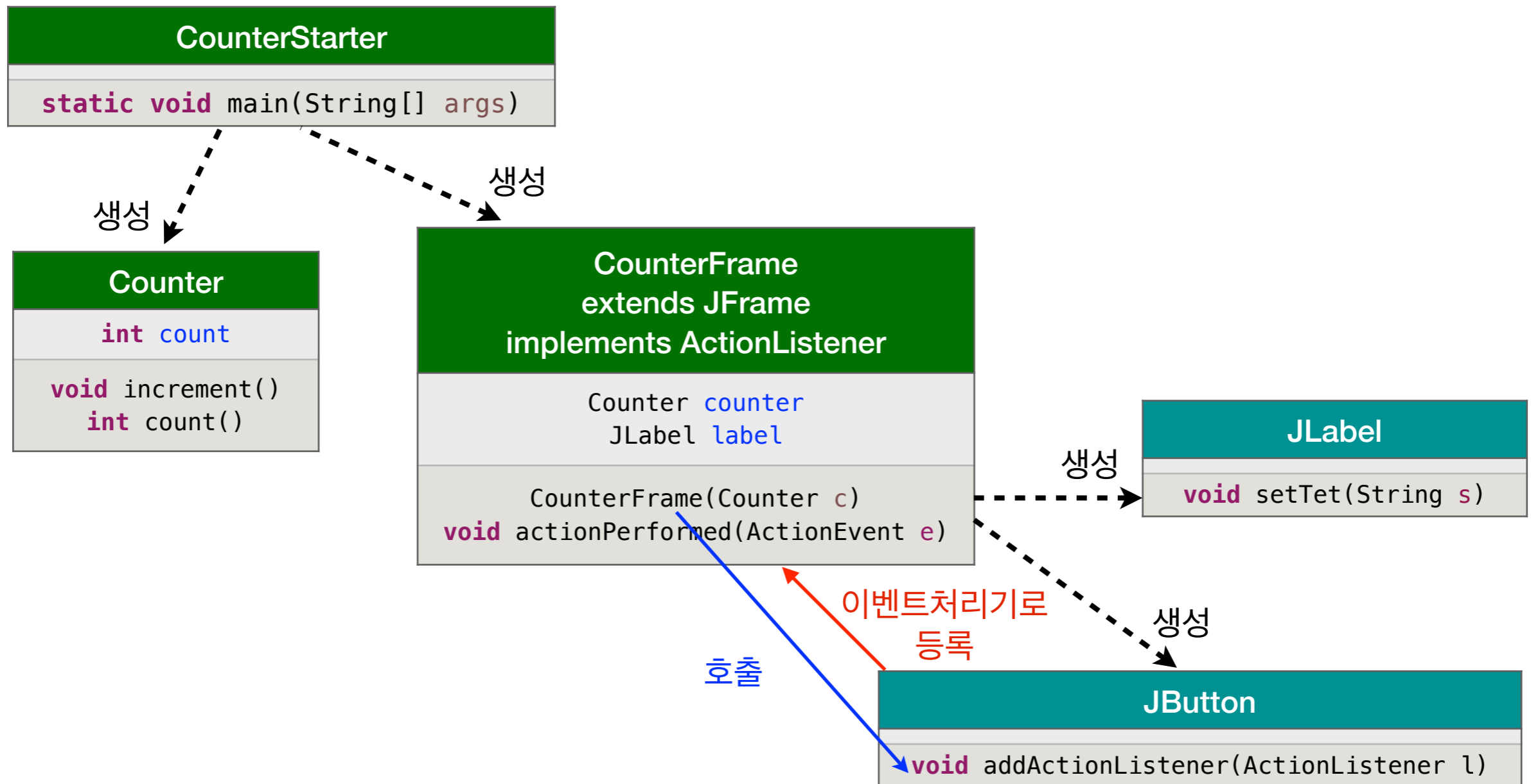


ActionEvent  
"button click"

actionPerformed

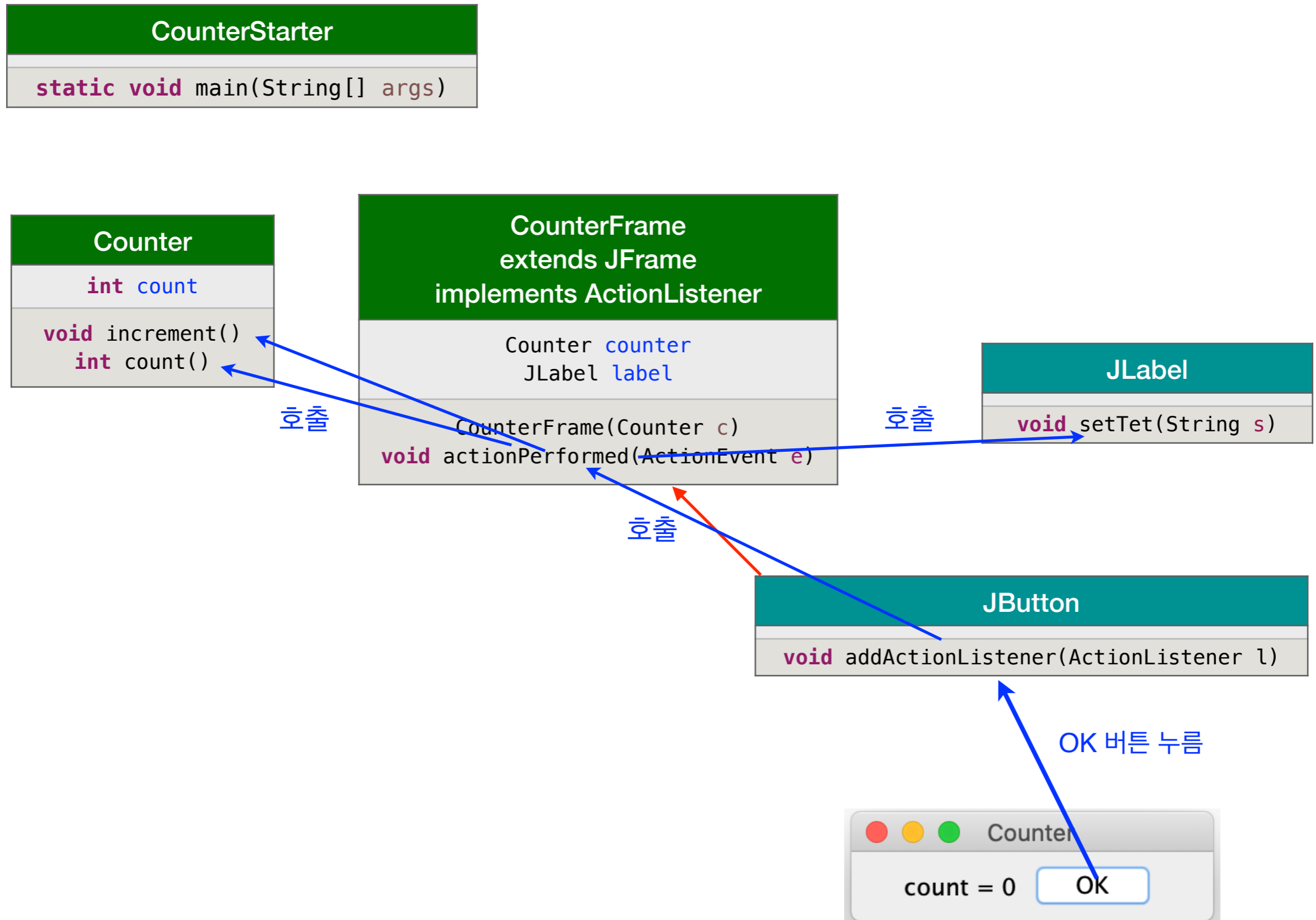
ActionListener

# 1 단계 : 초기 버전

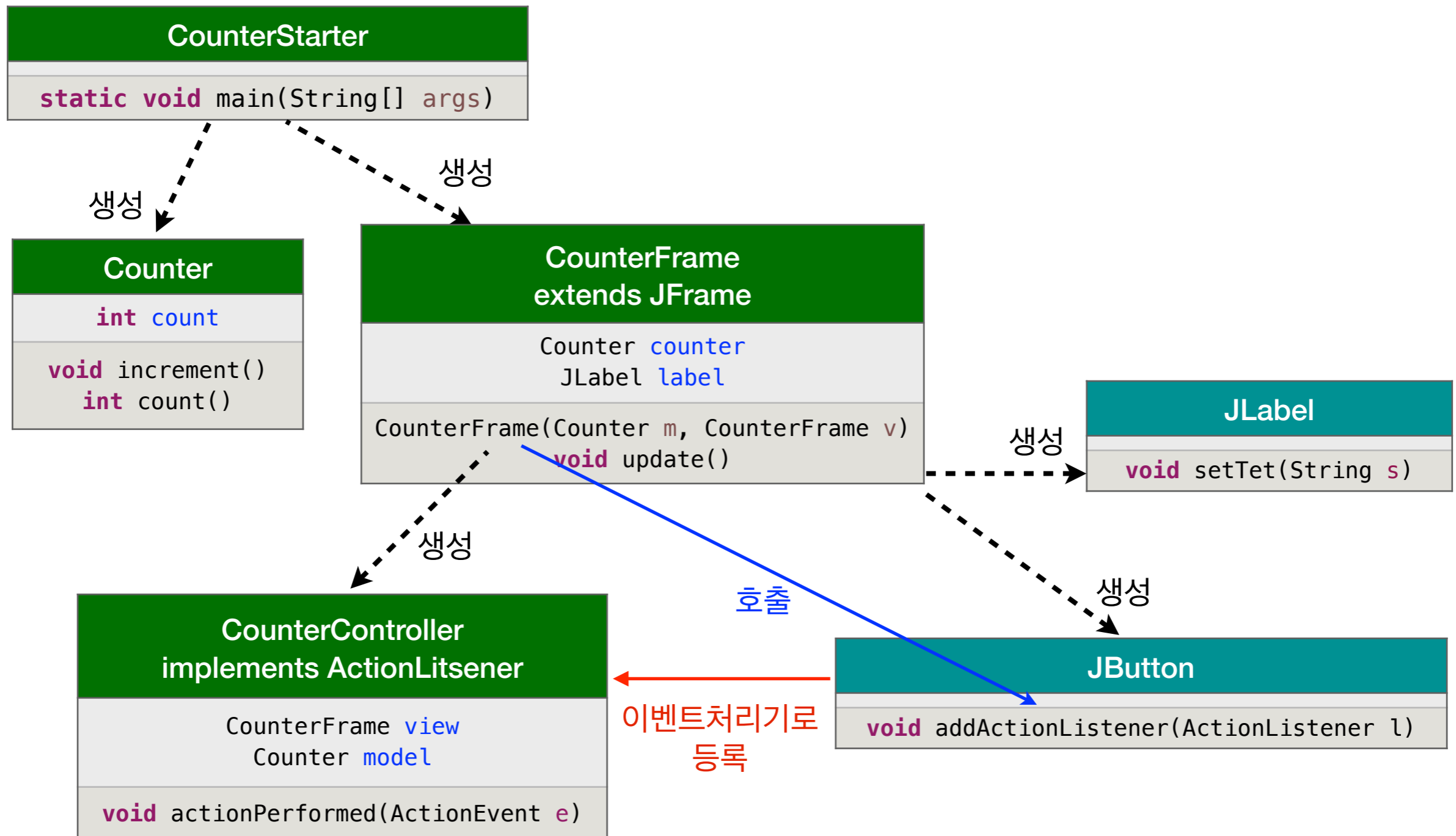




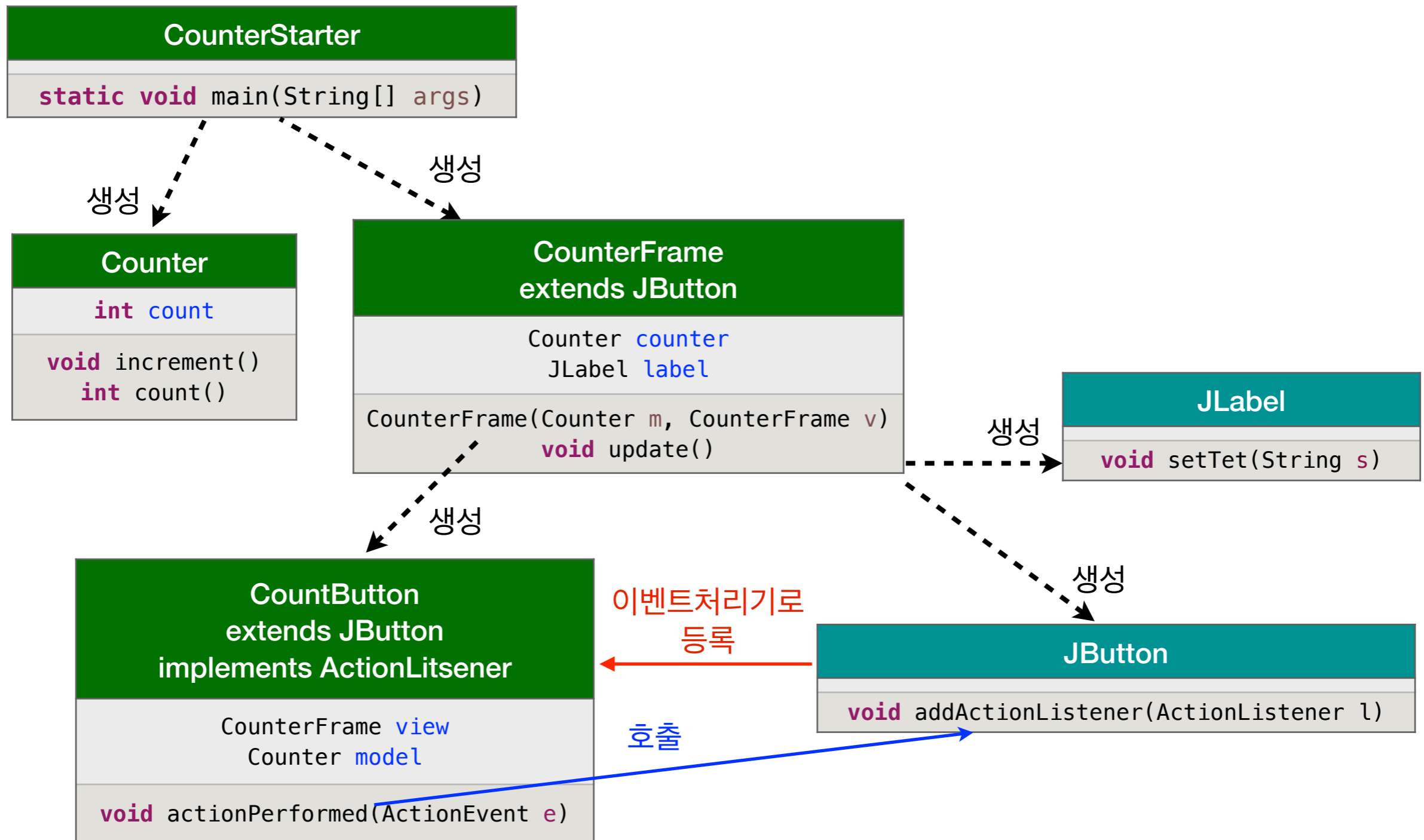
# 1 단계 : 초기 버전



## 2 단계 - 컨트롤러와 뷰를 분리



### 3 단계 - 버튼 전용 컨트롤러



# 4 단계 - 그래픽 카운터, 구역 정리, 버튼 추가

## Border Layout

